MedSparse Healthcare: A Sparse Mixture-of-Experts Medical Language Model Trained from Scratch on Consumer Hardware

Sudhanshu Shukla
Independent researcher, Bengaluru, India
https://su394611-del.github.io

Technical report · Version 0.9 (preprint), 23 November 2026
Stage 3 results will be added in version 1.0.

Abstract

We describe MedSparse Healthcare, a 38.8M-parameter decoder-only language model whose feed-forward layers are sparse mixtures of experts (four experts per layer, top-2 routing), trained from random initialization on healthcare text with a 30,001-token byte-level BPE tokenizer learned only from biomedical documents. Stage 1 pretrains on openly licensed PubMed full text from the Common Pile; Stage 2 applies full-parameter, completion-only supervised fine-tuning (SFT) on medical question–answer pairs. Evaluated with a float32 inference engine that runs in the browser, Stage 2 lowers held-out medical answer perplexity from 692.5 to 252.9 (ratio 0.365, 95% CI 0.343–0.389) while PubMed perplexity rises by 17% (ratio 1.166, 95% CI 1.087–1.244), and expert routing stays balanced (normalized routing entropy ≥ 0.966 in every layer). Multiple-choice accuracy on 150 MedMCQA validation questions (32.7% and 28.7%) is not distinguishable from the 25% chance level and stays below the 38% majority-answer baseline, which we report as a limitation of models at this scale. Stage 3 continues full-parameter SFT over all 5.8 million MIRIAD question–answer pairs, with a validation split that keeps whole source papers out of training, on a single consumer-grade GPU; it is in progress, and its results will be reported in version 1.0 of this report. We release the weights under the Apache License 2.0, together with the training code, the evaluation harness, and practical lessons for training on consumer hardware. The model is a research artifact and is not intended for clinical use.

1 Introduction

Most open medical language models are adapted from general-purpose models with billions of parameters, which places both their training and their inspection out of reach for individual researchers. This report takes the opposite route: a small sparse Mixture-of-Experts (MoE) model, built and trained end to end from random initialization, with every stage reproducible on modest hardware and every released checkpoint evaluated on held-out data with uncertainty estimates.

The architecture follows the sparse MoE decoder design popularized by Mixtral [6], scaled down by several orders of magnitude. MedSparse Healthcare is an independent implementation: it does not use or distribute Mistral AI or Mixtral weights and is not affiliated with Mistral AI.

Contributions.

- A from-scratch sparse MoE medical language model (38,779,008 parameters, 8.66M active non-embedding parameters per token) with a domain-specific tokenizer, released openly in two stages (pretrained and medical SFT).

Table 1: MedSparse Healthcare architecture.

Total parameters	38,779,008
Input + output embeddings	23,040,768 (59.4%)
Transformer blocks	15,737,856
Active non-embedding parameters per token	8,660,352
Layers / hidden size	4 / 384
Query heads / key-value heads	6 / 2 (GQA)
Experts per layer / experts per token	4 / 2
Expert intermediate size (SwiGLU)	768
Vocabulary	30,001 (byte-level BPE)
Position encoding	RoPE, $\theta = 10^6$
Normalization	RMSNorm, $\epsilon = 10^{-5}$
Maximum positions	2,048
Load-balancing loss coefficient	0.02

- A browser-based evaluation of both releases with bootstrap and Wilson 95% intervals, a paired comparison between stages, and per-layer routing statistics.
- Full-parameter SFT on the complete MIRIAD-5.8M corpus [19] on a single consumer-grade GPU, using a paper-disjoint validation split.
- Engineering lessons for long training runs on consumer hardware: data-pipeline bottlenecks, crash-safe checkpointing, and split-consistent evaluation.

2 Related work

Sparsely gated MoE layers route each token to a small subset of expert networks [15]; Switch Transformers simplified routing and introduced a load-balancing auxiliary loss [4], and Mixtral showed that top-2 routing over eight experts per layer yields strong open models [6]. In biomedicine, pretraining from scratch on in-domain text with an in-domain vocabulary has been shown to outperform continued pretraining of general models for encoders (PubMedBERT [5]) and generative models (BioGPT [9]). Recent medical instruction data include the verifiable-problem data and responses released with HuatuoGPT-o1 [3] and MIRIAD, 5.8 million question–answer pairs rephrased from and grounded in peer-reviewed literature [19]. MedMCQA provides more than 194k multiple-choice questions from Indian medical entrance exams [11].

3 Model

MedSparse Healthcare is a decoder-only Transformer [17] with pre-normalization (RMSNorm [18]), rotary position embeddings (RoPE [16]), grouped-query attention (GQA [1]) and a sparse MoE feed-forward block in every layer (Table 1). Each expert is a SwiGLU network [14]. A linear router produces a softmax over the four experts; the top two are selected per token and their weights renormalized to sum to one. A Switch-style auxiliary loss [4], $\mathcal{L}_{\text{aux}} = E \sum_{e=1}^{E} \bar{p}_e f_e$, where $\bar{p}_e$ is the mean router probability and f_e the fraction of top-2 assignments to expert e, is averaged over layers and added with coefficient 0.02. Input and output embeddings are untied.

At this scale the embeddings dominate the parameter count: the four Transformer blocks hold 15.7M parameters, of which each token activates 8.66M (attention plus two of four experts per layer).

4 Data and tokenizer

Tokenizer. A byte-level BPE tokenizer [12, 13] with special tokens `<pad>`, `<unk>`, `<bos>` and `<eos>` was trained from scratch on healthcare text, giving 30,001 tokens. The same tokenizer is used in every stage.

Pretraining corpus (Stage 1). PubMed full text from `common-pile/pubmed_filtered`, part of the Common Pile v0.1 [7], screened by two rules: the per-record license must be CC0, public domain or CC-BY 2.0–4.0, and the text must pass a clinical keyword screen. A deterministic document-hash partition holds out 2% of documents for validation.

Medical SFT data (Stage 2). The English configuration of `FreedomIntelligence/medical-o1-reasoning-SFT` [3], using the `Question` and `Response` fields only (the chain-of-thought field is not used as a target). A question-hash bucket holds out 5% for validation.

MIRIAD (Stage 3). MIRIAD-5.8M contains 5,821,948 question–answer pairs [19]. Because many pairs come from the same source paper, the validation split is drawn at the paper level. A pair is held out when

$$\mathrm{BLAKE2b}_{64}(\texttt{"42:"} \,\|\, \texttt{paper_id}) \,/\, 2^{64} < 0.02,$$

so no paper contributes to both training and validation.

Prompt format. All SFT stages use a fixed template with `### System`, `### User` and `### Assistant` sections. The loss is completion-only: system, user and assistant-prefix tokens are masked, and only answer tokens and the end-of-sequence token are supervised.

5 Training

Table 2 summarizes the three stages. All parameters (embeddings, attention, routers, experts, norms and the output head) are updated in every stage; no adapters or frozen layers are used. Optimization uses AdamW [8].

Stage 3 on a consumer-grade GPU. Full-parameter AdamW training of a 38.8M-parameter model needs roughly 16 bytes per parameter for weights, gradients and optimizer states (about 0.6 GB), so the full model, together with checkpointed activations, fits comfortably in the memory of a consumer-grade GPU. In Stage 3, when a question–answer pair does not fit in the 192-token window, the prompt keeps at most half of the window (its most recent tokens) and the answer is truncated with its end-of-sequence token retained. Data are streamed with a 1,024-example shuffle buffer. Every 500 optimizer steps, validation cross-entropy is measured on 64 held-out pairs and checkpoints are written. The run processes about 2.7–3.1 optimizer steps per second (11–12 examples per second), or roughly five to six days for one epoch; the exact wall-clock time will be reported with the Stage 3 results.

6 Evaluation

Protocol. Both released checkpoints were evaluated with the project's in-browser inference engine (JavaScript, float32, CPU, 8 threads; harness commit `0c5e96e`). Held-out sets are drawn with the training-time split functions, so no evaluation item was seen in training:

Table 2: Training stages. A dash marks a setting that was not recorded with the released checkpoint.

	Stage 1: pretraining	Stage 2: medical SFT	Stage 3: MIRIAD SFT
Initialization	Random	Stage 1 weights	Earlier MedSparse checkpoint
Data	PubMed (Common Pile, filtered)	medical-o1 (en)	MIRIAD-5.8M, paper-disjoint
Objective	Next token + aux. loss	Completion-only + aux. loss	Completion-only + aux. loss
Sequence length	128	128	192
Micro-batch × accumulation	—	1×4	1×4
Peak learning rate	—	2×10^{-5}	2×10^{-5}
Schedule	—	100 warm-up steps	100 warm-up steps, then constant
Weight decay / clipping	—	—	0.01 / 1.0
Optimizer steps	5,000	2,000	one epoch, ≈1.42M
Precision	—	—	fp32, gradient checkpointing
Hardware	—	—	Single consumer-grade GPU

Table 3: Held-out evaluation of the two released checkpoints (95% intervals in brackets). Uniform-guess perplexity is 30,001.

	Stage 1 (pretrained)	Stage 2 (medical SFT)
PubMed perplexity	572.15 [472.93, 704.14]	667.04 [546.16, 822.56]
PubMed top-1 next-token accuracy	15.6%	14.3%
Medical Q&A answer perplexity	692.47 [643.46, 740.98]	252.86 [233.14, 273.47]
Medical Q&A top-1 next-token accuracy	14.4%	21.6%
MedMCQA accuracy (length-normalized)	32.7% [25.7, 40.5]	28.7% [22.0, 36.4]
MedMCQA accuracy (sum of log-probs)	23.3% [17.3, 30.7]	23.3% [17.3, 30.7]
Decoding speed in the browser	13.7 tokens/s	14.0 tokens/s
Paired ratio, SFT / pretrained perplexity:		
Medical Q&A	0.365 [0.343, 0.389]	
PubMed	1.166 [1.087, 1.244]	

- **PubMed:** the first 40 articles of the first Common Pile PubMed shard that pass the training filters and fall in the 2% validation partition; first 129 tokens per article, 128 scored (5,120 tokens).
- **Medical Q&A:** 80 answers from the medical-o1 5% validation bucket, formatted exactly as in Stage 2 with completion-only scoring (7,943 answer tokens).
- **MedMCQA:** the first 150 validation questions with four non-empty options. Each option is scored by its mean log-probability per token after the question (cloze format; options are not shown in the prompt).

Perplexity intervals are 2,000-sample bootstrap intervals over items, token-weighted with a fixed seed; accuracy intervals are Wilson score intervals; the stage comparison is a paired bootstrap over the same items.

Results. Table 3 shows a clear specialization trade-off. Medical SFT reduces answer perplexity by 63.5% (ratio 0.365) and raises top-1 next-token accuracy on answers from 14.4% to 21.6%,

Table 4: Normalized routing entropy per layer on held-out PubMed and Q&A tokens (1.0 = perfectly even use of the four experts).

	Layer 1	Layer 2	Layer 3	Layer 4
Stage 1 (pretrained)	0.979	0.991	0.998	0.966
Stage 2 (medical SFT)	0.982	0.997	0.997	0.998

while perplexity on held-out PubMed text rises by 16.6%, consistent with partial forgetting of the pretraining distribution after 2,000 steps of narrow-domain SFT. MedMCQA accuracy is not separable from the 25% chance level for either checkpoint, and both are below the 38% obtained by always choosing the most frequent correct letter in this sample; length normalization alone moves accuracy from 23.3% to 28.7–32.7%. We therefore make no claim of exam-level medical knowledge at this scale.

Routing. Expert use stays balanced after SFT (Table 4): normalized routing entropy, $H/\log E$ over the share of top-2 selections per expert, is at least 0.966 in every layer of both checkpoints, and SFT raises it in three of four layers.

Stage 3 (MIRIAD). Stage 3 is in progress at the time of writing. Its paper-disjoint MIRIAD validation loss, and the metrics of Table 3 for the Stage 3 checkpoint measured with the same harness, will be added in version 1.0 of this report.

7 Engineering lessons from training on consumer hardware

The data pipeline, not the GPU, set the pace. Stage 3 throughput stayed near 12 examples per second whether the GPU processed one example or sixteen per step, because streaming, shuffling and tokenizing on the CPU could not supply examples faster. Pre-tokenizing the corpus into a memory-mapped array before training would remove this bottleneck; on a personal computer, other CPU load (for example a busy browser) measurably slows training.

Crash-safe checkpoints are essential for multi-day runs. An unexpected restart left an optimizer-state file that could no longer be loaded. Checkpoints are now written to a temporary name, flushed to disk with `fsync` and atomically renamed, and resuming selects the newest checkpoint whose files are all complete and verifies that data-order and optimization settings match the original run.

Evaluate with the exact training split. An early check of Stage 3 outputs drew "held-out" questions with a question-level rule instead of the paper-level rule used in training, so some evaluated questions came from training papers and the score was optimistic. All evaluation now imports the training split function itself.

Precision. Not every consumer GPU supports bfloat16. Mixed precision is possible with float16 autocast and dynamic loss scaling [10], but the reported Stage 3 run uses float32 for stability; gradient checkpointing [2] keeps activation memory small.

8 Limitations and responsible use

MedSparse Healthcare is a research model. It is not a medical device, has not been validated for clinical safety or accuracy, and must not be used for diagnosis, treatment, triage, medication dosing or patient communication. At 38.8M parameters it hallucinates and does not answer

exam questions above chance. The Stage 2 and Stage 3 corpora contain machine-generated text (MIRIAD pairs were produced with GPT-3.5-Turbo from literature passages [19]), which may carry errors or outdated guidance. Evaluation sets are small (40 articles, 80 answers, 150 questions), so intervals are wide; in the medical Q&A set only 8.8% of questions fit entirely within the 128-token window. The training data keep their own licenses, which users must check before reuse.

9 Availability

- Pretrained weights: `https://huggingface.co/sudhanshu42844/medsparse-healthcare-pretrained-from-scratch`
- Medical SFT weights: `https://huggingface.co/sudhanshu42844/medsparse-healthcare-medical-sft`
- Code, evaluation harness and results: `https://github.com/su394611-del/MedSparse-Healthcare-LLM-From-Scratch`
- Project site and in-browser playground: `https://su394611-del.github.io/MedSparse-Healthcare-LLM-From-Scratch/`

Weights and code are released under the Apache License 2.0 (copyright 2026 Sudhanshu Shukla). MedSparse Healthcare was formerly published as "Mixtral Healthcare"; the old repository and model URLs redirect to the new names.

References

[1] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. GQA: Training generalized multi-query Transformer models from multi-head checkpoints. In *Proceedings of EMNLP*, 2023. arXiv:2305.13245.

[2] T. Chen, B. Xu, C. Zhang, and C. Guestrin. Training deep nets with sublinear memory cost. arXiv:1604.06174, 2016.

[3] J. Chen, Z. Cai, K. Ji, X. Wang, W. Liu, R. Wang, J. Hou, and B. Wang. Towards medical complex reasoning with LLMs through medical verifiable problems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14552–14573, 2025. arXiv:2412.18925.

[4] W. Fedus, B. Zoph, and N. Shazeer. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[5] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1), 2021. arXiv:2007.15779.

[6] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, et al. Mixtral of experts. arXiv:2401.04088, 2024.

[7] N. Kandpal, B. Lester, C. Raffel, S. Majstorovic, S. Biderman, B. Abbasi, L. Soldaini, E. Shippole, et al. The Common Pile v0.1: An 8TB dataset of public domain and openly licensed text. In *Advances in Neural Information Processing Systems 38, Datasets and Benchmarks Track*, 2025. arXiv:2506.05209.

[8] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[9] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu. BioGPT: Generative pre-trained Transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409, 2022.

[10] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed precision training. In *International Conference on Learning Representations (ICLR)*, 2018.

[11] A. Pal, L. K. Umapathi, and M. Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, PMLR 174:248–260, 2022.

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. OpenAI technical report, 2019.

[13] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725, 2016.

[14] N. Shazeer. GLU variants improve Transformer. arXiv:2002.05202, 2020.

[15] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1701.06538.

[16] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. RoFormer: Enhanced Transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. arXiv:2104.09864.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, 2017.

[18] B. Zhang and R. Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32*, 2019.

[19] Q. Zheng, S. Abdullah, S. Rawal, C. Zakka, S. Ostmeier, M. Purk, E. Reis, E. J. Topol, J. Leskovec, and M. Moor. MIRIAD: Augmenting LLMs with millions of medical query-response pairs. arXiv:2506.06091, 2025.